

# TrajSV: A Trajectory-based Model for Sports Video Representations and Applications

Zheng Wang\*, Shihao Xu\*, Wei Shi†

**Abstract**—Sports analytics has received significant attention from both academia and industry in recent years. Despite the growing interest and efforts in this field, several issues remain unresolved, including (1) data unavailability, (2) lack of an effective trajectory-based framework, and (3) requirement for sufficient supervision labels. In this paper, we present TrajSV, a trajectory-based framework that addresses various issues in existing studies. TrajSV comprises three components: data preprocessing, Clip Representation Network (CRNet), and Video Representation Network (VRNet). The data preprocessing module extracts player and ball trajectories from sports broadcast videos. CRNet utilizes a trajectory-enhanced Transformer module to learn clip representations based on these trajectories. Additionally, VRNet learns video representations by aggregating clip representations and visual features with an encoder-decoder architecture. Finally, a triple contrastive loss is introduced to optimize both video and clip representations in an unsupervised manner. The experiments are conducted on three broadcast video datasets to verify the effectiveness of TrajSV for three types of sports (i.e., soccer, basketball, and volleyball) with three downstream applications (i.e., sports video retrieval, action spotting, and video captioning). The results demonstrate that TrajSV achieves state-of-the-art performance in sports video retrieval, showcasing a nearly 70% improvement. It outperforms baselines in action spotting, achieving state-of-the-art results in 9 out of 17 action categories, and demonstrates a nearly 20% improvement in video captioning. Additionally, we introduce a deployed system along with the three applications based on TrajSV.

**Index Terms**—sports video representation, triple contrastive learning, trajectory

## I. INTRODUCTION

WITH the advancements in high-resolution cameras and sensors, significant amounts of data can now be gathered from sports events and subjected to analysis. This data analysis holds the potential to uncover valuable insights for a broad range of sports application scenarios, including coaching [1], [2], [3], [4], [5] and broadcasting [6], [7], [8], [9]. Although some sports analytic techniques have been developed for the aforementioned applications, they still suffer from some of the issues among (1) data unavailability, (2) lack of an effective trajectory-based framework, and (3) requirement of sufficient supervision labels, as summarized in Table I. Next, we discuss these issues and present our proposals for solving them.

**(1) Data unavailability.** Many sports-related studies [1], [2], [5] require the collection of player or ball movement data (i.e., trajectories) using professional equipment like GPS

TABLE I  
THE ISSUES IN EXISTING STUDIES (✓ INDICATES THE ISSUE EXISTS AND × OTHERWISE). ISSUE-1: DATA UNAVAILABILITY, ISSUE-2: LACK OF AN EFFECTIVE TRAJECTORY-BASED FRAMEWORK, AND ISSUE-3: REQUIREMENT OF SUFFICIENT SUPERVISION LABELS.

	Issue-1	Issue-2	Issue-3
play2vec [1]	✓	×	×
Chalkboard [2]	✓	✓	×
LearnRank [5]	✓	×	×
SimScene [6]	×	✓	✓
LRML [10]	×	✓	✓
CALF [9]	×	✓	✓
LRRCN [11]	×	✓	✓
Baidu-VC [12]	×	✓	✓
VMCG [13]	×	✓	✓

trackers or optical tracking cameras, which can be expensive. For example, studies on sports play retrieval [1], [2], [5], [14] utilize 4K resolution cameras in an optical tracking system, costing thousands of dollars for tracking a single game [15], not including the expense of a professional team to operate it. Additionally, the collected data is limited to specific sports games (e.g., France’s Ligue de Football Professionnel (LFP) in these studies), making it challenging to generalize the research results to games without available movement data. This highlights the need for more cost-effective and accessible methods of collecting sports data. We propose to use publicly available sports broadcast videos as an alternative data source. With the widespread availability of sports broadcasts on the internet, this approach has the potential to provide a large volume of data at a relatively low cost.

**(2) Lack of an effective trajectory-based framework.** Learning sports data representation is a widely used technique with applications in sports video retrieval [6], [10], action spotting [9], [11], and video captioning [12], [13]. However, few studies have explored a trajectory-based framework that can support different downstream tasks. In many cases, existing techniques focus on utilizing data for specific tasks, like sports video retrieval, often overlooking the underlying structure and information embedded in trajectories, which could enhance flexibility for various tasks. To address this gap, we investigate a trajectory-based framework that extracts spatio-temporal trajectories from raw sports broadcast videos, deriving both video-level and clip-level representations. These representations support a variety of downstream tasks and can be fine-tuned for improved results. The rationale of extracting trajectories is that it embeds both spatial and temporal features of a sports game, providing a shared foundation for different tasks. For example, the trajectory of a ball or player can be used for recognizing specific actions (action spotting) and serve as information for retrieving similar scenes based on extracted trajectory patterns (sports retrieval). Recognizing these

\*Equal contribution.

†Wei Shi is the corresponding author.

Zheng Wang, Shihao Xu and Wei Shi are with Huawei Technologies, Co., Ltd. E-mail: {wangzheng155, shihao.xu, w.shi}@huawei.com

common patterns behind trajectories presents an opportunity to develop techniques for various sports applications. We note that some recent video models (e.g., VideoMAE [16] or X-CLIP [17]) have shown advancements in addressing open-world applications. However, we empirically observe that these models do not perform very well when applied to sports, a domain highly related to player movements, exhibiting unique characteristics not present in other video types.

**(3) Requirement of sufficient supervision labels.** Several existing studies [6], [10] have been developed to learn sports video representations under a supervised learning framework. However, creating labeled video datasets for supervised learning can be a time-consuming and expensive process. For example, in studies on soccer scene retrieval [6], [10], authors manually annotate video footage to identify soccer events such as shots or corner kicks. The resulting dataset is limited to a set of specific events, and its effectiveness heavily relies on sufficient labeled data. The limitation highlights the need of developing an unsupervised learning framework that utilizes the inherent structure of sports videos and identifies trajectory patterns without explicit labels. Unsupervised learning offers a more flexible way to learning representations.

**Summary of our solution.** In this paper, we develop a trajectory-based framework for sports video representations, called TrajSV, which avoids the aforementioned three issues. For (1), TrajSV differs from existing techniques [1], [2], [5], [14] that rely on limited spatial data for training and evaluation. In contrast, TrajSV takes advantage of raw broadcast videos available on the Web, which significantly expands the accessibility and diversity of data. To achieve this, TrajSV incorporates a comprehensive data preprocessing pipeline, including video segmentation, camera calibration, and multi-object tracking. For (2), TrajSV focuses on task-agnostic representations for various sports analytic tasks that rely on trajectories, which provides new insight for continued research in this line of studies. To achieve this, TrajSV introduces CRNet and VRNet. In CRNet, the model effectively captures sequential data (e.g., videos), and combines visual and spatial features to derive the clip representations. Additionally, VRNet incorporates an attention mechanism implemented by two attention blocks (called MAB and MSB) via an encoder-decoder architecture. Our solution addresses the issue of uneven clip contribution to video representation by prioritizing the most important clips. This leads to an overall improvement in the video representation's quality. For (3), we present a novel approach called triple contrastive learning, which optimizes both video and clip representations in an unsupervised manner by comparing their similarities. The rationale behind this approach is the relative ease of identifying the more similar video among two candidates when a query video is provided. To further enhance the comparison between videos, we introduce a triple contrastive loss that takes into account three aspects: 1) trajectory patterns within clips, 2) dependencies of clips in a video, 3) and mutual information derived from different variants of the same video.

**Novelty.** We discuss several novel aspects of the TrajSV design as follows. 1) Utilizing Raw Broadcast Videos: TrajSV is built on raw broadcast videos available on the Web, which

overcomes the limitation of sports movement data availability. With a larger and diverse dataset, TrajSV improves the performance of the model. 2) Task-Agnostic Representations: TrajSV focuses on trajectory-based representations in sports videos, allowing their versatile use in various sports analytic tasks that depend on trajectories. This enhances flexibility in sports video analysis. 3) The Architecture: TrajSV incorporates three key components, including data preprocessing, CRNet, and VRNet. The incorporation of both clip-level and video-level representations in one framework is a unique feature not found in previous studies. 4) Triple Contrastive Learning: it uniquely considers three contrasts of its kind. First, contrasting complex movement patterns in sports videos helps in understanding actions and events within each clip. Second, contrasting dependencies between different clips within a video helps understand temporal continuity and context. Third, contrasting variants of the same video captures diverse cues, enhancing robust representation.

Our contributions can be summarized as follows.

- We propose TrajSV, a trajectory-based framework for learning sports video representations. Our model addresses three issues in this field: 1) it utilizes raw broadcast videos, making it widely applicable, 2) it learns task-agnostic representations that can be used for different applications, and 3) it does not require supervision labels.
- We present a deployed system (used in a real sports video search engine) based on TrajSV, and three applications built on top of the system: 1) sports video retrieval, 2) action spotting, and 3) video captioning, which correspond to using the representations at both video-level (i.e., 1) and clip-level (i.e., 2 and 3). To the best of our knowledge, this is the first industry system of its kind.
- We evaluate the TrajSV on *three* broadcast datasets with various baselines, and the results demonstrate its effectiveness across *three* types of sports (i.e., soccer, basketball, and volleyball), for the *three* applications. The results demonstrate TrajSV's state-of-the-art performance in sports video retrieval, with an improvement of nearly 70%. Additionally, it improves baselines in action spotting, achieving state-of-the-art results in 9 out of 17 action categories, and in video captioning, exhibiting an improvement of nearly 20%.

## II. RELATED WORK

### A. Sports Data Analytics

We review the literature on sports data analytics in terms of different data types, i.e., spatio-temporal data and video data. For spatio-temporal data, it is to track the moving objects in a sports game with some professional devices, e.g., optical tracking cameras have been installed in a sports field to track the trajectories of players and the ball in a soccer game [1]. The data attracts many research efforts [1], [14], [2], [5], [3], [4], [18], [19]. For example, play2vec [1] learns a representation based on spatio-temporal sports trajectories using a Seq2Seq architecture. Chalkboard [2] is also based on trajectory inputs and adopts a role-based pairwise matching strategy to compute the similarity for the retrieval task. Sports-Traj [19] advances

sports analytics by introducing a unified model for trajectory prediction, imputation, and spatio-temporal recovery. It extends Mamba [20] with a bidirectional temporal design and integrates a Transformer encoder to enhance temporal feature extraction. Overall, this line of research is based on spatio-temporal data tracked from professional cameras, and this makes it difficult to apply these studies to sports games where the tracking data is not available. In this work, we explore alternative data sources (i.e., broadcast videos) that have better availability on the Web, to support a wider range of tasks.

For video data, the sports analytic tasks include sports video retrieval [6], [10], [7], [8], [21], action spotting [9], [22], [23], [24], [25], [26], [27], video captioning [12], camera shot segmentation [28], [29], camera calibration [30], [31], tracking, re-identification [32], [33], [34], and summarization [35]. The work [36] provides a detailed evaluation of these tasks based on their collected datasets. For example, SimScene [6], [10] learns video clip representations based on human annotations of some common soccer scenes (e.g., shot, corner kick), where it utilizes a BiLSTM-based model and incorporates multimodal features (e.g., images, audio, and text) into the representation. SportSense [8] is a system that supports interactive sports video retrieval, where a user can freely sketch a path on the soccer field, and it returns video clips that match the sketched path. Further, an auto-suggest feature is deployed into the system, where it suggests potential directions when the user sketches a path, and thus it relaxes the use of system memory and increases the retrieval speed [21]. The action spotting task [37], entails pinpointing both the timing and type of a particular action within a video, where each action is annotated with a single timestamp. CALF [9] employs a context-aware loss function that emphasizes the temporal context surrounding each action, as opposed to solely focusing on a single annotated frame for spotting. LRRCN [11] presents a joint framework for athlete tracking and action recognition in sports videos, featuring a robust tracker for precise athlete localization and a long-term recurrent convolutional network for modeling temporal action cues. The papers [25], [38] provide extensive surveys on video action spotting across over 10 sports, examining datasets, methodologies, and applications. Con-RPM [26] is a self-supervised model for group activity representation that captures evolving interactions in team sports using predictive coding. It integrates individual and scene context via a Transformer-based encoder-decoder, optimized with contrastive and adversarial learning. KARI [27] enhances group activity recognition by leveraging concretized knowledge from training data. It models action co-occurrence and spatial distributions as structured maps to improve relation inference and individual representation. The video captioning task [12], centers on generating textual comments anchored with single timestamps. This involves a two-stage process. First, a spotting model combines frame features into a single clip representation, which is then used to generate proposal timestamps. Further, these timestamps are subsequently utilized by the captioning model to produce the anchored comments. VCMG [13] introduces a hierarchical neural network with motion representation and group relationship modules to improve sports video captioning. Our work is different from

these studies in two aspects. First, we learn representations based on raw broadcast videos instead of those short clips, where we derive both video-level and clip-level representations to support downstream tasks. Nevertheless, this is not the focus of existing studies. Second, in contrast to the studies [6], [7], our model is trained with an unsupervised learning framework (i.e., triple contrastive learning) without human annotations.

### B. Video Representation Learning

Video representation learning is a fundamental task that aims to capture informative representations from video data, enabling various downstream applications such as action recognition [39], [40], [41], and video captioning [40]. For example, Qian et al. [39] learn video representations for human action recognition, it introduces a Dense Predictive Coding (DPC) framework, which enables the learning of dense spatio-temporal representations from videos by predicting future representations. Recently, contrastive learning, e.g., video-text alignment [42], [17], [43], [44], [45] has also been widely used to learn video representations. For example, X-CLIP [17] improves the video-text retrieval task using a cross-frame communication transformer and a multi-frame integration transformer, and maximizes the similarity between the paired video and text. In this work, we develop a trajectory-based framework that combines scene-based video representations to learn sports video representations, which can be trained by comparing similar sports videos with dissimilar ones in an unsupervised way.

## III. PRELIMINARIES AND PROBLEM STATEMENT

**Sports Video.** Let  $V$  denote a sports video, representing an untrimmed broadcast video such as a soccer game on platforms like YouTube. The video  $V$  consists of a sequence of images, i.e.,  $V = \langle I_1, I_2, \dots, I_{|V|} \rangle$ , where  $I_i$  denotes an image at the  $i^{th}$  frame.

**Sports Clip.** A sports clip  $C$  refers to a portion of  $V$ . Due to the nature of broadcast videos, we focus clips that are sports-related (e.g., a clip that contains offensive movements which lead to a goal), but not those that are sports-unrelated (e.g., a video clip that contains a dance performance at the opening of a soccer game).

**Trajectory.** A trajectory  $T$  corresponds to a sequence of locations, i.e.,  $(x_1, y_1), (x_2, y_2), \dots, (x_{|T|}, y_{|T|})$ , that captures the movement of an object (e.g., players or the ball) in a sports clip, where  $(x_i, y_i)$  denotes the location tracked from the image at the  $i^{th}$  frame. Therefore, a sports clip that contains multiple moving objects such as players can be represented by a set of multiple trajectories.

**Problem Statement.** Given a sports video  $V$ , we aim to, 1) learn a vector representation  $\mathbf{v}$  for the  $V$ , and 2) derive the representation  $\mathbf{c}_i$  ( $1 \leq i \leq n$ ) for any sports clip  $C_i$  contained in the video  $V$ ,  $n$  denotes the number of clips in the  $V$ .

Note that our goal is to learn representations of sports videos at both the video-level and the clip-level, such that the derived representations could be utilized in various video-based applications (e.g., sports video retrieval [6]) and clip-based

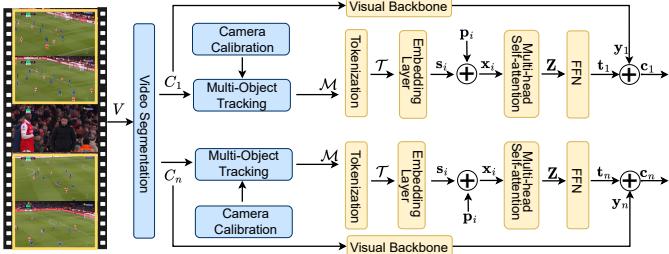


Fig. 1. The CRNet framework. A raw broadcast video  $V$  undergoes video segmentation to obtain multiple clips  $C_1, \dots, C_n$ . For each clip, the underlying camera parameters are estimated, and these parameters are then utilized in a multi-object tracking process to extract trajectories. Then, these extracted trajectories are represented as a sequence of segment matrices  $\mathcal{M}$ , which are further processed through a Transformer encoder to obtain a representation  $t$ . The clip representation  $c$  is generated by concatenating (denoted by  $\oplus$ ) the  $t$  with a visual-based representation  $y$ .

applications (e.g., action spotting [9], video captioning [12]), respectively.

#### IV. MODEL ARCHITECTURE

TrajSV presents a trajectory-based framework for sports video representations. It consists of three components, namely data preprocessing (Section IV-A), CRNet (Section IV-B) and VRNet (Section IV-C). We investigate a triple contrastive learning method to train the TrajSV in an unsupervised manner (Section IV-D).

##### A. Data Preprocessing

Raw sports videos consist of sequences of images, often containing irrelevant content and lacking spatial features crucial for player and ball movement analysis. To address these challenges, the data preprocessing involves three steps: (1) extracting sports clips, (2) extracting camera parameters from video images, and (3) tracking player and ball trajectories by mapping image coordinates to a sports field coordinate system using the obtained camera parameters for each clip.

**Video Segmentation.** A key step in raw video editing is segmenting a long video into multiple clips. This task is modeled as a classification problem [36], where each video frame is classified into predefined camera types (e.g., main camera center, close-up player). A segmented clip consists of consecutive frames with the same type. Our approach adopts a ResNet-based method [9], sequentially classifying the camera type at each frame based on its extracted ResNet features. Sports clips correspond to classes with specific camera types (e.g., main camera center, left and right). If a clip is highly fragmented (duration below a threshold, which is set to 0.4s based on empirical findings), it is removed from the outputs.

**Camera Calibration.** It is a crucial step in mapping trajectory points from pixel coordinates in a video image to world coordinates, such as a sports field. This process involves estimating camera parameters to achieve accurate mapping. Our calibration algorithm utilizes field markings, such as lines and circle segments in a soccer field, as reference points, following [30]. The algorithm minimizes reprojection errors from pixel coordinates to world coordinates, calculated as the

Euclidean distance between pixels detected by localization algorithms (e.g., DeepLabV3 [46]) and the reprojected field markings with known coordinates in a given sports field.

**Multi-Object Tracking (MOT).** MOT involves extracting trajectories of multiple objects in a video, utilizing a multi-task approach [47] that includes object detection and re-identification. In object detection, the model locates object positions in each video frame using bounding boxes, outputting center coordinates. These coordinates are mapped to sports field coordinates using camera parameters from calibration. The model optimizes three parallel objectives [48]: heatmaps (for object center locations), object center offsets (for precise localization), and bounding box sizes (for height and width estimation). For re-identification (re-ID), the model recognizes the same object across different camera views. ResNet extracts re-ID features from bounding boxes, and the model classifies each box, determining whether to link the object to an existing track or create a new track.

**Real-world Data Processing Challenges.** We discuss the real-world challenges in data processing and our engineering efforts to support the reproduction of TrajSV. (1) Handling Sports-Unrelated Content in Broadcasts: In real-world sports videos, non-sports-related content is often prevalent. To address this, we study video segmentation techniques to classify different camera types (e.g., main camera center, left, and right) to separate relevant sports content from irrelevant footage. (2) Camera Calibration for Sports Videos: Sports videos, especially those captured from various angles, require precise camera calibration. We use field markings, such as lines and circle segments on a soccer field, as reference points for calibration to ensure accurate spatial representation in the video frames. (3) Multi-Object Tracking for Key Sports Elements: Tracking critical elements like the ball is essential for sports analysis. Since the ball plays a central role in many sports, we fine-tune multi-object tracking (MOT) models on a sports-specific dataset. Additionally, we empirically set a weight parameter (i.e., 10) for smaller objects to enhance the tracking performance, particularly for the soccer ball. These engineering efforts are essential for ensuring that TrajSV can be reproduced and applied effectively in real-world scenarios.

##### B. Clip Representation Network (CRNet)

In CRNet, it aims to learn the representations of clips (i.e., a set of tracked trajectories). We design a trajectory-enhanced Transformer module, by fusing visual features (extracted from images) and spatial features (extracted from trajectories) together in a clip. Figure 1 illustrates the architecture.

**Tokenization.** For each clip, we first break it into a sequence of non-overlapping segments with a fixed duration, and each is called a segment. To tokenize the trajectories in each segment, we then partition the sports field into a grid with equal cell size, and each segment corresponds to a binary matrix, called segment matrix (denoted by  $\mathcal{M}$ ), where we set the cells (corresponding entries in the matrix) to 1 if they are traveled through by the trajectories, and 0 otherwise. This strategy provides a controllable resolution of sports scenes via the cell

size. With a smaller one, it has a higher resolution but reduces the robustness against errors of tracked trajectory locations, and vice versa. Furthermore, we map the segment matrices into tokens (denoted by  $\mathcal{T}$ ) by following [1]. The core idea is to scan the segment matrices one by one, and for each matrix, we create a new token to represent it if it is dissimilar (measured by the Jaccard index) from those matrices that have been scanned.

**Input Embedding Layer.** We transform the extracted segment matrices into real vectors via an embedding layer. The Transformer [49], which has received great success in modeling sequential data, is used for this purpose. In contrast to recurrent neural networks (RNN) that receive inputs sequentially, the Transformer-based model in CRNet uses a self-attention mechanism that operates on all input tokens in parallel. Thus, it is insensitive to the order of inputs. To preserve the sequential information of trajectories in a clip, we construct the input representations as follows:

$$\mathbf{x}_i = \mathbf{s}_i + \mathbf{p}_i, \quad (1)$$

where  $\mathbf{s}_i$  and  $\mathbf{p}_i$  denote an input segment embedding and a learnable positional embedding at the  $i^{th}$  segment, respectively. The positional embeddings enable the model to be aware of the order of segments rather than treating them as an unordered set.

**Multi-head Self-attention.** To model dependencies between the video segments in a clip, we use a multi-head self-attention mechanism [50] based on the input representations  $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m] \in \mathbb{R}^{m \times d_1}$ , where  $m$  denotes the number of segments, and each is with the dimension  $d_1$ . The multi-head self-attention mechanism maps the input representations to output representations  $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_m] \in \mathbb{R}^{m \times d_2}$ , where  $d_2$  is the dimension of each output segment. The mechanism is formulated as follows:

$$\mathbf{Z} = \text{Multi-head}(\mathbf{X}, \mathbf{X}, \mathbf{X}) = \text{Concat}(O_1, O_2, \dots, O_h) \mathbf{W}^O, \quad (2)$$

$$O_i = \text{Attention}(\mathbf{X} \mathbf{W}_i^Q, \mathbf{X} \mathbf{W}_i^K, \mathbf{X} \mathbf{W}_i^V) \quad (3)$$

$$= \delta\left(\frac{(\mathbf{X} \mathbf{W}_i^Q)(\mathbf{X} \mathbf{W}_i^K)^\top}{\sqrt{d_1/h}}\right) \mathbf{X} \mathbf{W}_i^V, \quad (4)$$

where the three positions in the multi-head correspond to query, key, and value. The input representations are projected into  $h$  subspaces (called heads), which allow the model to jointly attend to information from the independent head  $O_i$  ( $1 \leq i \leq h$ ) by concatenating them together, and  $\mathbf{W}^O \in \mathbb{R}^{d_1 \times d_2}$  denotes the weight of the multi-head. Then, self-attention is applied to each head, where  $\mathbf{W}_i^Q$ ,  $\mathbf{W}_i^K$ ,  $\mathbf{W}_i^V \in \mathbb{R}^{d_1 \times d_1/h}$  denote the query, key, and value transformations, respectively. The softmax function  $\delta(\cdot)$  is applied.

**Position-wise Feed-forward Network.** The output representation  $\mathbf{Z}$  is then fed into a fully connected network with two dense layers, i.e., Position-wise Feed-forward Network. It consists of two linear transformations with a ReLU activation function  $\Phi(\cdot)$  in between, that is

$$\mathbf{t} = \Phi(\mathbf{Z} \mathbf{W}_1 + \mathbf{b}_1) \mathbf{W}_2 + \mathbf{b}_2, \quad (5)$$

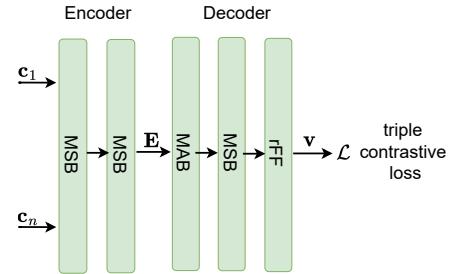


Fig. 2. The VRNet framework. It encodes the clip representations  $\mathbf{c}$  into features  $\mathbf{E}$ , which are then decoded to produce the final video representation  $\mathbf{v}$ . Finally, a triple contrastive loss  $\mathcal{L}$  is introduced to optimize these representations in an unsupervised manner.

where  $\mathbf{t} \in \mathbb{R}^{d_3}$  denotes the trajectory-based representation, and  $\mathbf{W}_1, \mathbf{W}_2, \mathbf{b}_1, \mathbf{b}_2$  are the parameters of the network.

**Fusion of Trajectory and Visual Representations.** Apart from capturing the sports trajectory information, we also leverage a video representation backbone [17] to extract visual information as another modality to further boost representation performance. The video representation model can capture visual information across frames and convert frames into a fixed-length vector, denoted as  $\mathbf{y} \in \mathbb{R}^{d_4}$ . Finally, the derived trajectory-based representation is concatenated with the visual-based representation to represent a sports clip, which is formulated as:

$$\mathbf{c} = \text{Concat}(\mathbf{t}, \mathbf{y}), \quad (6)$$

where  $\mathbf{c} \in \mathbb{R}^{d_5}$  ( $d_5 = d_3 + d_4$ ) denotes the output clip representation.

### C. Video Representation Network (VRNet)

In VRNet, it learns a video representation  $\mathbf{v}$  based on the derived clip representations  $\mathbf{C} = [\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_n] \in \mathbb{R}^{n \times d_5}$ , where  $n$  represents the number of clips contained in the video. To do this, one immediate idea is to aggregate the clip representations as the video representation. However, this design assumes that each clip contributes equally to the final representation, whereas it is more intuitive to assign more weight to those highlighted clips (e.g., a clip that contains a goal). Inspired by this, we design two attention blocks (called MAB and MSB) on those clips in VRNet, and the video representation is outputted via an encoder-decoder-based architecture. Figure 2 illustrates the architecture.

**Encoder.** We first define a Multihead Attention Block (MAB) as a building block in the encoder, as follows:

$$\text{MAB}(\mathbf{X}, \mathbf{Y}) = \text{LayerNorm}(\mathbf{H} + \text{rFF}(\mathbf{H})), \quad (7)$$

$$\mathbf{H} = \text{LayerNorm}(\mathbf{X} + \text{Multi-head}(\mathbf{X}, \mathbf{Y}, \mathbf{Y})), \quad (8)$$

where  $\mathbf{X}$  and  $\mathbf{Y}$  denote two input representations to be pairwise attended by multi-head.  $\text{rFF}(\cdot)$  denotes a row-wise feed-forward layer that operates each instance independently and returns the same shape as the input.  $\text{LayerNorm}(\cdot)$  denotes a layer normalization operation [51]. Further, when  $\mathbf{X} = \mathbf{Y}$ , the MAB reduces to a self-attention building block called Multihead Self-attention Block (MSB), that is

$$\text{MSB}(\mathbf{X}) = \text{MAB}(\mathbf{X}, \mathbf{X}) \quad (9)$$

Based on MSB, we construct an encoder as follows:

$$\mathbf{E} = \text{Encoder}(\mathbf{C}) = \text{MSB}(\text{MSB}(\mathbf{C})). \quad (10)$$

The objective of MSB is to transform clip representations  $\mathbf{C} \in \mathbb{R}^{n \times d_5}$  into features  $\mathbf{E} \in \mathbb{R}^{n \times d_6}$  and capture the correlation of those clips by self-attention in between.

**Decoder.** The decoder further aggregates the features  $\mathbf{E}$  into a single video representation  $\mathbf{v} \in \mathbb{R}^d$ , which is formulated as

$$\mathbf{v} = \text{Decoder}(\mathbf{E}) = \text{rFF}(\text{MSB}(\text{MAB}(\mathbf{s}, \mathbf{E}))), \quad (11)$$

The objective of MAB is to utilize a trainable seed vector  $\mathbf{s} \in \mathbb{R}^d$  to reduce the  $\mathbf{E}$  into a single representation by attention, and then feeds into a feed-forward network to output the final video representation with the same shape as  $\mathbf{s}$ .

#### D. Optimization

To learn the video representations, we present triple contrastive learning. It helps to train the model in an unsupervised manner by introducing comparison. Specifically, let  $\mathcal{V}^{(1)}$  denote a batch of sports videos and for each video denoted by  $V^{(1)}$  contained in  $\mathcal{V}^{(1)}$ , it corresponds to a vector representation  $\mathbf{v}_i^{(1)} (1 \leq i \leq n)$ , where  $n$  denotes the batch size. Then, we introduce two variants of  $V^{(1)}$  (resp.  $\mathcal{V}^{(1)}$ ,  $\mathbf{v}^{(1)}$ ), denoted by  $V^{(2)}$  (resp.  $\mathcal{V}^{(2)}$ ,  $\mathbf{v}^{(2)}$ ) representing intra-clip videos and  $V^{(3)}$  (resp.  $\mathcal{V}^{(3)}$ ,  $\mathbf{v}^{(3)}$ ) representing inter-clip videos, as follows.

**Intra-clip Video.** Recall that a clip corresponds to a set of multiple trajectories, where we randomly replace some of the trajectories with other trajectories from the whole training set. Here, we control the replacement with a noise rate denoted by  $\delta$  from 0 to 1, e.g., by setting a larger ratio, it tends to replace more trajectories, and thus the variant  $V^{(2)}$  will be more dissimilar to the original  $V^{(1)}$  in terms of the trajectory patterns within the clips.

**Inter-clip Video.** Each video corresponds to a set of clips after data prepossessing. Similarly, we replace some of the clips with other clips sampling from the training set controlled by  $\delta$ . In this way, the variant  $V^{(3)}$  is dissimilar to the original  $V^{(1)}$  in terms of the clip dependencies in the video.

**Triple Contrastive Loss.** We further introduce a triple contrastive loss  $\mathcal{L}$  based on  $V^{(1)}$ ,  $V^{(2)}$ , and  $V^{(3)}$ , where it contains three contrasts:  $\mathcal{L}(V^{(1)}, V^{(2)})$ ,  $\mathcal{L}(V^{(1)}, V^{(3)})$  and  $\mathcal{L}(V^{(2)}, V^{(3)})$ . To calculate  $\mathcal{L}(V^{(1)}, V^{(2)})$ , we denote  $V^{(2)}$  as the positive of  $V^{(1)}$ , and we consider the negatives as those videos in the same batch with  $V^{(1)}$ . Here, we note that the training data is randomly shuffled, and the negatives within a batch are generally quite dissimilar to the query. We denote a symmetric InfoNCE loss for  $\mathcal{L}(V^{(1)}, V^{(2)})$ , which is composed of two terms  $\mathcal{L}_{1,2}$  and  $\mathcal{L}_{2,1}$ . For  $\mathcal{L}_{1,2}$ , it is defined as:

$$\mathcal{L}_{1,2} = \sum_{V_i^{(1)} \in \mathcal{V}^{(1)}} -\log \frac{\exp(\mathbf{v}_i^{(1)} \cdot \mathbf{v}_i^{(2)}/\tau)}{\sum_{V_j^{(2)} \in \mathcal{V}^{(2)}, j \neq i} \exp(\mathbf{v}_i^{(1)} \cdot \mathbf{v}_j^{(2)}/\tau)}, \quad (12)$$

where  $\tau$  denotes a temperature parameter in contrastive learning. Symmetrically, we can get  $\mathcal{L}_{2,1}$  by anchoring at  $V^{(2)}$ , then  $\mathcal{L}(V^{(1)}, V^{(2)})$  is defined as

$$\mathcal{L}(V^{(1)}, V^{(2)}) = \mathcal{L}_{1,2} + \mathcal{L}_{2,1}. \quad (13)$$

Similarly, we can calculate  $\mathcal{L}(V^{(1)}, V^{(3)})$  and  $\mathcal{L}(V^{(2)}, V^{(3)})$ , and the triple contrastive loss  $\mathcal{L}$  is defined as

$$\begin{aligned} \mathcal{L} = & \alpha \mathcal{L}(V^{(1)}, V^{(2)}) + \beta \mathcal{L}(V^{(1)}, V^{(3)}) \\ & + (1 - \alpha - \beta) \mathcal{L}(V^{(2)}, V^{(3)}), \end{aligned} \quad (14)$$

where  $0 \leq \alpha, \beta \leq 1$  are hyperparameters that are used to balance the importance of each contrast.

We discuss the rationale behind the design. First, inspired by the fact that discerning the similarity between two videos can be intricate. However, given a query video and two candidate videos, it is easier to tell the model which one (i.e., positive sample) is more similar than the other one (i.e., negative sample) to the query, and therefore we can optimize the representations by comparing the similarities of videos in this unsupervised manner. Second, we enhance the representation by contrasting the videos  $V^{(2)}$  and  $V^{(3)}$  with the original  $V^{(1)}$ . The contrasts help the representation to capture both trajectory patterns within the clips and their dependencies in a video. Additionally, contrasting  $V^{(2)}$  and  $V^{(3)}$  further enhances the representation by maximizing the mutual information derived from the two variants of the same video  $V^{(1)}$ .

## V. DEPLOYMENT AND APPLICATIONS

### A. Deployment

A sports system, built upon TrajSV, is integrated into a practical sports video search engine. The system architecture is illustrated in Figure 3. For offline use, the TrajSV model is employed to extract representations from all database videos, forming an Approximate Nearest Neighbor (ANN) index (specifically, HNSW [52]). This index facilitates rapid retrieval by computing vector similarities. To optimize memory usage and efficiency, compression and quantization techniques [53] are applied, trading off some accuracy. For online use, a query video undergoes TrajSV to derive clip representations from CRNet and a video representation from VRNet. After compression and quantization, the video representation is used for similarity retrieval through the ANN index. By specifying a Top- $K$  parameter, result videos from the database are retrieved and recommended to users. The system is extended to enhance action spotting and video captioning by utilizing clip representations from CRNet.

### B. Applications

We introduce the following three applications developed on top of the system: sports video retrieval, action spotting, and video captioning.

**Sports Video Retrieval.** Sports video retrieval is an emerging application used in several sports broadcast platforms, such as ESPN, to recommend videos to sports fans based on their interests. The retrieval process involves the offline stage and

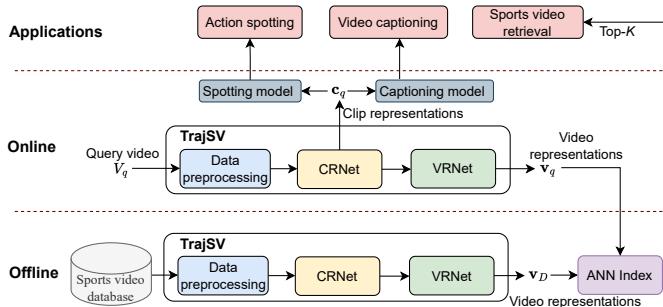


Fig. 3. System deployment on TrajSV.

online stage as introduced in the deployed system, where it uses the video representation for the retrieval.

**Action Spotting.** The action spotting task, as defined in [37], involves localizing when and which a given action (e.g., goal, shots on target) occurs in a video, with each action annotated with a single timestamp. To support this task, we utilize clip representations as a plug-in built on top of existing spotting models. We achieve this by concatenating the representations with the embedded features of the model, creating a new input. We then unfreeze the representations, allowing them to be further optimized with the model parameters. This is because the clip representations embed spatial features from extracted trajectories that are potentially associated with the actions to be detected. For example, the CALF model [9] leverages ResNet to extract per-frame feature representations. We concatenate the representation of a clip where the frame falls with the embedded features, which creates a new input representation that is then fed into the model for fine-tuning.

**Video Captioning.** We consider two video captioning tasks that are introduced in Baidu-VC [12], called Single-anchored Dense Video Captioning (SDVC) and Dense Video Captioning (DVC). For SDVC, the task involves analyzing a video, identifying specific moments that require captions (Commentary Spotting), and generating natural language sentences that describe the events taking place at those moments. To do this, we follow a similar approach in the action spotting task. We concatenate the clip representation with the embedded moment feature, creating a new input representation to fine-tune the baseline models (e.g., Baidu-VC) for video captioning. For DVC, the task involves temporally localizing a caption with the start and end frames instead of anchoring a single moment.

## VI. EXPERIMENTS

### A. Experimental Setup

**Datasets and Ground Truth.** We conduct experiments on three broadcast sports video datasets, i.e., YouTube, SoccerNet [36], and SportsMOT (it is employed for retrieval and qualitative experiments, lacking labels necessary for evaluating action spotting and video captioning tasks) [55]. For YouTube, we crawl 3,261 soccer videos from YouTube sports channels with durations varying from 13 to 962 seconds. The SoccerNet dataset contains 550 complete broadcast soccer games from the major European leagues and provides annotations for action spotting in 17 common classes (e.g., goal, offside,

shots on target). Additionally, the dataset contains an average of 78.33 comments per game that are temporally localized, corresponding to a total of 36,894 captions for the entire dataset. For SportsMOT, it consists of 240 videos across three types of sports (soccer, basketball, and volleyball), where the videos are collected from some main sports events like the Olympic Games, NCAA Championship, and NBA. To track trajectories, we set the sports field coordinates to a range of  $[-52.5, +52.5]$  meters along the x-axis and  $[-34, +34]$  meters along the y-axis, following the setting described in [1], the setting is also used for non-soccer data (i.e., basketball and volleyball).

We then discuss the ground truth for evaluation. For (1) sports video retrieval, we introduce variations in the original videos contained in the datasets by manipulating them with different noise rates  $\delta$ . This manipulation involves replacing both trajectories and clips in the videos. Subsequently, we use the manipulated videos as queries and aim to retrieve their corresponding original versions (i.e., ground truth) from the database. For (2) action spotting and (3) video captioning, we take the common classes and the comments provided by the SoccerNet dataset as the ground truth, respectively.

**Baselines.** We carefully examine the literature including a recent challenge paper [56], and identify recent baselines in terms of (1) sports video retrieval, i.e., play2vec [1], Chalkboard [2], SimScene [6], ResNet [54], X-CLIP [17], (2) action spotting, i.e., CALF [9], NetVLAD++ [22], Baidu-AS [24], and (3) video captioning, i.e., Baidu-VC [12]. The descriptions of these methods can be found in Section II. Notably, these models are open-sourced, and we tune their parameters to the best for the comparison.

**Implementation Details.** We conduct the experiments using Python 3.7 and PyTorch 1.8.0. We describe the implementation details with open-sourced codes below.

(1) For data preprocessing, in video segmentation, a 1D CNN with three layers and a 21-frame kernel is trained for binary classification to identify the main-camera view. The model extracts ResNet 512-dimensional PCA-reduced features<sup>2</sup>, aiding in segmenting long videos into shorter clips. We adopt morphological opening and closure techniques [57] with a kernel size of three to enhance binary prediction quality. In camera calibration, following TVCalib [30]<sup>3</sup>, a segment localization model is trained for instance segmentation (e.g., lines and circle segments in a soccer field). TVCalib predicts camera and distortion parameters by minimizing the reprojection loss. Following the original hyperparameter setup, we use AdamW [58] with a learning rate of 0.05 and weight decay of 0.01 to train the camera parameters over 2000 fixed steps, employing the one-cycle learning rate schedule [59] with  $pct_{start} = 0.5$ . Batch size and downsample Frames per Second (FPS) are set to 512 and 5, respectively. In multi-object tracking, we fine-tune the FairMOT model [47]<sup>4</sup> on the SoccerNet-Tracking dataset [60] for 40 epochs. The model uses a variant of DLA-34 [61] as the backbone. Training

<sup>2</sup><https://github.com/SoccerNet/sn-spotting>

<sup>3</sup><https://github.com/mm4spa/tvcalib>

<sup>4</sup><https://github.com/ifzhang/FairMOT>

<sup>1</sup>All results are statistically significant (t-test with  $p < 0.05$ ).

TABLE II

EFFECTIVENESS OF SPORTS VIDEO RETRIEVAL (HR@1 AND MRR) ON YOUTUBE, SOCCERNET, AND SPORTSMOT (SOCCER, BASKETBALL, AND VOLLEYBALL). THE BEST RESULTS ARE MARKED IN BOLD, AND THE SECOND-BEST RESULTS ARE UNDERLINED<sup>1</sup>.

Method	YouTube						SoccerNet						SportsMOT					
	HR@1			MRR			HR@1			MRR			HR@1			MRR		
	0.5	0.55	0.6	0.5	0.55	0.6	0.5	0.55	0.6	0.5	0.55	0.6	0.5	0.55	0.6	0.5	0.55	0.6
play2vec(Mean) [1]	0.069	0.075	0.031	0.105	0.111	0.058	0.078	0.031	0.058	0.133	0.090	0.052	0.097	0.106	0.057	0.071	0.063	0.031
play2vec(MLP) [1]	0.223	0.205	0.088	0.331	0.299	0.156	0.312	0.203	0.156	0.443	0.322	0.221	0.328	0.301	0.154	0.210	0.232	0.084
Chalkboard [2]	0.077	0.000	0.000	0.109	0.019	0.010	0.300	0.280	0.140	0.372	0.357	0.242	0.326	0.218	0.224	0.167	0.250	0.194
SimScene (Mean) [6]	0.129	0.141	0.038	0.199	0.209	0.087	0.180	0.080	0.080	0.285	0.138	0.168	0.276	0.254	0.121	0.170	0.191	0.061
SimScene (MLP) [6]	0.191	0.170	0.061	0.276	0.254	0.121	0.120	0.060	0.040	0.219	0.102	0.098	0.007	0.011	0.004	0.003	0.007	0.001
ResNet (Mean) [54]	0.347	0.364	0.130	0.452	0.467	0.238	0.438	0.156	0.078	0.571	0.328	0.222	0.454	0.379	0.099	0.250	0.281	0.031
ResNet (MLP) [54]	0.659	0.658	0.231	0.779	0.779	0.443	0.609	0.281	0.109	0.762	0.500	0.325	0.711	0.672	0.347	0.438	0.531	0.125
X-CLIP (Mean) [17]	0.358	0.380	0.119	0.506	0.521	0.237	0.562	0.234	0.094	0.699	0.431	0.302	0.665	0.680	0.282	0.469	0.500	0.156
X-CLIP (MLP) [17]	0.611	0.611	0.216	0.768	0.764	0.426	0.641	0.312	0.141	0.762	0.538	0.332	0.604	0.527	0.117	0.281	0.406	0.062
TrajSV	<b>0.791</b>	<b>0.802</b>	<b>0.475</b>	<b>0.881</b>	<b>0.887</b>	<b>0.680</b>	<b>0.719</b>	<b>0.484</b>	<b>0.250</b>	<b>0.846</b>	<b>0.689</b>	<b>0.551</b>	<b>0.748</b>	<b>0.810</b>	<b>0.614</b>	<b>0.712</b>	<b>0.650</b>	<b>0.475</b>

TABLE III  
EFFECTIVENESS OF ACTION SPOTTING (AVG-MAP%) ON SOCCERNET<sup>1</sup>.

Method	Overall	Ball out	Throw-in	Foul	Ind. free-kick	Clearance	Shots on tar.	Shots off tar.	Corner	Substitution	Kick-off	Yellow card	Offside	Dir. free-kick	Goal	Penalty	Yel. → Red	Red card
# of instances	22551	6460	3809	2414	2283	1631	1175	1058	999	579	514	431	416	382	337	41	14	8
CALF [9]	40.7	63.9	56.4	53.0	41.5	51.6	26.6	27.3	71.8	47.3	37.2	41.7	25.7	43.5	72.2	30.6	0.7	0.7
NetVLAD++ [22]	53.4	70.3	69.0	64.2	44.4	57.0	39.3	41.0	79.7	68.7	62.1	56.7	39.3	57.8	71.6	79.3	3.7	4.0
Baidu-AS [24]	73.2	87.5	87.5	84.1	70.9	76.7	<b>66.3</b>	<b>66.7</b>	89.0	76.3	82.7	<b>77.9</b>	68.8	74.9	87.3	82.7	39.8	<b>25.9</b>
Baidu-AS+play2vec	73.5	87.4	<b>87.7</b>	84.0	71.2	76.6	65.9	66.3	89.0	78.4	82.4	77.3	<b>69.0</b>	<b>75.2</b>	87.5	82.4	<b>44.5</b>	24.1
Baidu-AS+TrajSV	<b>73.7</b>	<b>87.6</b>	87.6	<b>84.2</b>	<b>71.8</b>	<b>76.7</b>	65.3	66.3	<b>89.0</b>	<b>79.4</b>	<b>82.8</b>	77.6	68.5	75.1	<b>87.6</b>	<b>82.8</b>	44.2	25.8

is conducted using the Adam optimizer [62] with an initial learning rate of  $10^{-4}$ , which decays to  $10^{-5}$  after 20 epochs. We use a batch size of 12. A weight parameter of 10 is assigned to small objects (bounding box size  $< 500$  pixels) to improve soccer ball tracking. The fine-tuned FairMOT model is then used to track players and balls in all main-camera video clips, maintaining the original sampling frequency by default. After this, we transform the tracking result from the image coordinate into the real-world coordinate (sports field) by using the result of the camera calibration and MOT. Meanwhile, we filter out the objects that are tracked outside the sports field.

(2) For CRNet, following [1]<sup>5</sup>, we segment the sports field with a three-meter cell size and a one-second time segment length to tokenize trajectories. We set the number of video segments in a clip to 16 with zero padding ( $m = 16$ ), and randomly choose 16 consecutive clips for each video ( $n = 16$ ). The segment matrices are then tokenized with a Jaccard threshold of 0.3. The dimensions of the embedding layer, segment embedding, and transformer encoder are all set to 128 ( $d_1 = d_2 = d_3 = 128$ ). The dropout value in the positional encoding layer is 0.3, and the transformer encoder employs two layers and two heads. For visual representation, X-CLIP [17] is used as the backbone (pre-trained on Kinetics-600 dataset)<sup>6</sup>, representing each video clip as a 512-dimensional vector ( $d_4 = 512$ ). Visual and trajectory vectors are concatenated into a single blend vector, representing the video clip with a dimension of 640 ( $d_5 = d_3 + d_4 = 640$ ).

(3) For VRNet, the encoder has two MSB layers that transform the input dimension from 640 to 1280, and the decoder subsequently reduces the encoded embedding to the output dimension ( $d_6 = 128$ ). The attention model employs two heads.

(4) For training, the dataset is divided into 80% for training and 20% for testing. The model is trained for 100 epochs using the triple contrastive loss. A noise rate, randomly sampled between 0 and 0.2, generates intra-clip and inter-clip videos. The rationale is twofold. 1) We adopt random sampling within a controlled range to generate diverse intra-clip and inter-clip variants, enriching the training signal for contrastive learning. 2) Empirical tuning shows that setting  $\delta > 0.2$  introduces excessive noise, likely disrupting trajectory or clip coherence and degrading the model's ability to learn consistent representations. The chosen range thus achieves a favorable trade-off between sample diversity and training stability. The hyperparameters for  $\alpha$  and  $\beta$  in the loss function are set to 0.5 and 0.3, respectively. The early stopping with a patience step is set to 10. The learning rate is 0.01, and stochastic gradient descent (SGD) with a momentum factor of 0.7 serves as the optimizer. The temperature parameter is adjusted to 0.1.

**Evaluation Metrics.** We evaluate the effectiveness of TrajSV in terms of different tasks. (1) For sports video retrieval, Hitting Ratio for Top-1 (denoted by HR@1) and Mean Reciprocal Rank (denoted by MRR) are used by following [63], [64]. (2) For action spotting, Avg-mAP ( $\delta$  is varied from 5 to 60 seconds) is used by following [9], [22], [23], [37], [36]. (3) For video captioning, METEOR, BLEU, ROUGE, CIDEr, and SODA\_c are used by following [12]. Overall, a higher evaluation metric (i.e., HR@1, MRR, Avg-mAP, METEOR,

<sup>5</sup><https://github.com/zhenwang125/play2vec>

<sup>6</sup><https://github.com/microsoft/VideoX/tree/master/X-CLIP>

BLEU, ROUGE, CIDEr, SODA\_c) indicates a better result.

### B. Experimental Results

**(1) Sports Video Retrieval.** To evaluate performance, we introduce variations ( $V_+$ ) in original videos ( $V$ ) by applying noise rates from 0.5 to 0.6. If  $V_+$  is the top-1 retrieved video,  $\text{HR}@1=1$ ; otherwise,  $\text{HR}@1=0$ . MRR is defined as  $\text{MRR} = \frac{1}{\text{rank}}$ , where the  $\text{rank}$  denotes the rank of  $V_+$  in the database. The average results of HR@1 and MRR are reported in Table II. We observe that our TrajSV outperforms the best baseline methods significantly. For example, on YouTube (resp. SoccerNet and SportsMOT) with a noise rate of 0.6, TrajSV outperforms ResNet (MLP) (resp. X-CLIP (MLP) and ResNet (MLP)) by 105.6% (resp. 77.3% and 76.9%), because the fusion of visual and trajectory information enhances retrieval performance, surpassing single-modality approaches. TrajSV shows a similar trend on SportsMOT (covering soccer, basketball, and volleyball), highlighting its effectiveness in handling diverse sports videos.

**(2) Action Spotting.** As shown in Table III, we incorporate our clip-level representations with the Baidu-AS embeddings to enhance action spotting on SoccerNet. We observe an overall improvement in action spotting Avg-mAP by 0.5%, showcasing the effectiveness of including self-trained embeddings in identifying sports actions. Compared to Baidu-AS and Baidu-AS+play2vec (merging Baidu-AS embeddings with play2vec embeddings), our approach achieves state-of-the-art results in 9 out of 17 action categories. Notable improvements include 11.1% for yellow to red cards, 4.1% for substitution, and 1.3% for indirect free-kicks. We note that although the improvement over Baidu-AS+Play2Vec is small, it is still appreciated, as Baidu-AS has already reached the dataset's bottleneck performance by effectively capturing action spotting with vision-based features. The extra gain from incorporating trajectory features (e.g., TrajSV) suggests that vision cues dominate the task, but the added trajectory information remains valuable for refining performance.

**(3) Video Captioning.** Following [12], we evaluate video captioning using our clip-level representations combined with the Baidu-VC encoder on SoccerNet, and the results are reported in Table IV. TrajSV achieves state-of-the-art performance on most metrics, with significant improvements of 7.4% and 5.6% in mAP@30 and mAP@60 for the commentary spotting task, respectively. The video captioning performance also shows relative improvements up to 20.5% (R@30) on the DVC and SDVC tasks. Additionally, our results outperform the implementation that concatenates Baidu-VC embeddings with play2vec embeddings. These findings indicate that incorporating trajectory-informative representations enhances performance for commentary spotting and SDVC.

**(4) Ablation Study.** To evaluate the effectiveness of different components in TrajSV, we conduct an ablation study for sports video retrieval on YouTube as shown in Table V, where we (1) replace the CRNet with LSTM and BiLSTM layers, and the VRNet with mean pooling and a two-layer MLP; (2) test the performance of utilizing different features, including trajectory-based, X-CLIP, and ResNet features; (3) evaluate the

effect of the triple contrastive loss. For (1), it demonstrates that both the CRNet and VRNet components contribute to improving the performance. When the CRNet (resp. VRNet) is replaced with BiLSTM and LSTM (resp. Mean and MLP), the HR@1 decreases by 11.8% and 19.2% (resp. 49.7% and 29.5%), respectively. For (2), our observations reveal that the integration of both trajectory and visual representations yields superior results compared to using either representation individually. Specifically, excluding trajectory features leads to a 49.1% drop in HR@1 with X-CLIP features and 18.7% with ResNet features, highlighting the importance of trajectory-based representations in sports video retrieval. When combined with visual features, the trajectory-visual fusion further enhances fine-grained visual information across frames, leading to a 27.6% improvement in HR@1. For (3), we compare the performance when excluding each of the three components in the loss calculation, which shows that using triple contrastive loss is better than using either of them individually.

**(5) Parameter Study (batch size, cell size, and embedding dimension).** We investigate the impact of varying (1) batch size, (2) cell size, and (3) the dimension of trajectory representations for sports video retrieval. The results are presented in Table VI, Table VII, and Table VIII, respectively. For (1), the best results are achieved with a batch size of 128. This aligns with the expectation that a larger batch size typically improves performance in contrastive learning by incorporating more negative samples. For (2), the best results are achieved with a cell size of 3 meters, which balances the risk of multiple trajectories being encoded to the same representation (if the cell size is too large) and the risk of representing similar trajectories differently (if the cell size is too small). For (3), we find that increasing the dimension of trajectory representations improves retrieval performance. When the embedding dimension reaches 256, the retrieval performance HR@1 is 0.564. It suggests that a higher dimensional representation tends to preserve more information, which often results in better performance.

**(6) Transferability Study and of Generalization Discussion.** We evaluate the transferability of TrajSV for sports video retrieval on YouTube using SoccerNet as the training data. The transferability is studied in two ways: 1) training the model on SoccerNet and directly testing on YouTube (zero-shot), and 2) training on SoccerNet, fine-tuning using the YouTube training set, and further testing on YouTube. For comparison, we present results using ResNet features, and both training and testing on the YouTube dataset. As shown in Table IX, fine-tuning outperforms zero-shot for both TrajSV and ResNet. Fine-tuning with TrajSV is superior to ResNet, but zero-shot transfer lacks the same improvement, potentially attributed to differences in trajectory distributions between the two datasets. Moreover, fine-tuning performs worse than direct training on YouTube, likely due to differences in video length and content.

We further discuss the generalization capability of TrajSV from two perspectives. (1) Cross-Dataset Transferability: TrajSV can be easily transferred between datasets for the same sport (e.g., soccer). As shown in Table IX, it outperforms state-of-the-art methods (e.g., ResNet) and can achieve further

TABLE IV  
EFFECTIVENESS OF VIDEO CAPTIONING (BLEU, METEOR, ROUGE, CIDER, SODA\_C) ON SOCCERNET<sup>1</sup>.

Method	Commentary Spotting (mAP@ (%))			Dense Video Captioning (DVC)				Single-anchored Dense Video Captioning (SDVC)				
	5	30	60	B@4	M	R	C	B@4@30	M@30	R@30	C@30	SODA_C
Baidu-VC [12]	5.27	49.40	63.10	6.62	23.84	24.53	21.45	21.63	29.44	22.09	27.20	7.79
Baidu-VC+Play2Vec	<b>6.07</b>	47.70	60.97	6.48	23.83	24.26	20.58	20.94	21.93	26.40	27.11	7.73
Baidu-VC+TrajSV	4.00	<b>53.07</b>	<b>66.64</b>	<b>6.78</b>	<b>24.26</b>	<b>24.80</b>	<b>22.34</b>	20.35	21.97	<b>26.61</b>	<b>27.38</b>	<b>7.90</b>

TABLE V

ABLATION STUDY ON YOUTUBE<sup>1</sup>.

Components	HR@1	MRR
CRNet + VRNet	<b>0.475</b>	<b>0.680</b>
w/o CRNet (BiLSTM)	0.419	0.641
w/o CRNet (LSTM)	0.384	0.625
w/o VRNet (Mean)	0.239	0.147
w/o VRNet (MLP)	0.335	0.197
Trajectory + VRNet	0.344	0.444
Visual (X-CLIP) + VRNet	0.242	0.461
Visual (ResNet) + VRNet	0.386	0.603
w/o $\mathcal{V}^{(1)}$ in loss	0.448	0.662
w/o $\mathcal{V}^{(2)}$ in loss	0.433	0.646
w/o $\mathcal{V}^{(3)}$ in loss	0.441	0.637

TABLE VI

PARAMETER STUDY OF BATCH SIZE ON YOUTUBE.

Parameter	Value	Training time (min)	HR@1	MRR
Batch size	32	60	0.442	0.647
	64	42	0.440	0.646
	96	36	0.456	0.658
	128	25	<b>0.475</b>	<b>0.680</b>

improvements with additional training. (2) Cross-Sport Adaptability: The design of TrajSV is adaptable to other sports (e.g., basketball and volleyball), as demonstrated in Table II. This suggests that trajectory features provide a shared foundation across different sports videos, as they inherently capture both spatial and temporal information.

**(7) Scalability Study.** We conduct a scalability test based on the setup for sports video retrieval, where we vary the database size from 500 to 3,000 videos. We extract representations for all videos in the database and use the HNSW index to manage these representations. As shown in Table X, TrajSV demonstrates good scalability, which aligns with the expected vector search scalability complexity. It is important to note that the main computational complexity comes from embedding the given query sports videos, which involves necessary attention operations. This embedding occurs only once per query video and can then be used for search across a large database. Additionally, the embedding supports other analytical tasks, such as action spotting and video captioning.

**(8) Qualitative Results.** To qualitatively validate the effectiveness, we present some typical cases of (1) sports video retrieval and (2) video captioning in Figure 4 and Figure 5, respectively. For (1), we query a video clip, and then return the Top-1 retrieved clip with TrajSV. We note that the retrieved clip generally aligns with the query, featuring same sports actions such as corners or direct free-kicks. This observation extends to volleyball and basketball as well. Additionally, TrajSV incorporates trajectory (temporal) information alongside the visual features captured by X-CLIP [17]. To further explore the contribution of trajectory information, we illustrate the Top-1 retrieved clip using X-CLIP for corner and direct free-

TABLE VII  
PARAMETER STUDY OF CELL SIZE ON YOUTUBE.

Parameter	Value	HR@1	MRR
Cell size	1	0.469	0.678
	3	<b>0.475</b>	<b>0.680</b>
	5	0.467	0.673
	7	0.458	0.662
	9	0.433	0.653

TABLE VIII  
PARAMETER STUDY OF EMBEDDING DIMENSION ON YOUTUBE.

Parameter	Value	HR@1	MRR
Embedding dimension	32	0.436	0.665
	64	0.464	0.682
	128	0.497	0.697
	256	<b>0.564</b>	<b>0.738</b>

kick queries in Figure 4. We observe that the clips returned by TrajSV better align with the player movement patterns in the queries, suggesting that trajectory information is crucial for capturing similar movement patterns that visual features alone may not adequately represent. For (2), we present a case study of the SDVC, where we present the ground truth and the captions generated by the Baidu-VC, Baidu-VC + play2vec embeddings, and Baidu-VC + TrajSV embeddings. We observe that Baidu-VC + TrajSV achieves more accurate captions in content and timing. However, both the Baidu-VC and Baidu-VC + play2vec produce a different caption, possibly from previous time steps.

**(9) Comparison with General Video Models.** We compare recent state-of-the-art general video models to justify the necessity of the trajectory-specific design in TrajSV for sports analytics. In particular, we adopt StreamMind [65], a recent LLM-based model trained on various video understanding tasks. Notably, the SoccerNet dataset [36] is included in its pre-training. To represent a video, we average the frame-level embeddings produced by StreamMind. As shown in Table XI, TrajSV outperforms StreamMind by 10.8% in HR@1 and 6.7% in MRR. This gain is largely attributed to the incorporation of trajectory features. When these features are excluded, the performance of TrajSV drops—by 16.8% in HR@1 and 11.3% in MRR. We also observe that StreamMind slightly outperforms TrajSV without trajectory features, likely due to its large-scale LLM-based pre-training.

**(10) Model Generalization Across Diverse Sports Scenarios.** We present a detailed breakdown of performance across various sports scenarios on the SoccerNet dataset [36], which provides rich annotations to support scenario-specific analysis. As shown in Table XII, we report results in terms of MRR with a noise rate of  $\delta = 0.6$ , covering three categories: 1) Concrete events, including free kicks (both indirect and direct), corners, goals, and penalties. 2) Abstract patterns, involving transition-related events such as throw-ins and kick-



Fig. 4. Qualitative results of sports video retrieval, illustrating improvements from incorporating trajectory information in corner and direct free-kick queries.

TABLE IX  
TRANSFERABILITY ON YOUTUBE (YT) AND SOCCERNET (SN).

Method	Train → Test	Mode	HR@1				
			0.4	0.45	0.5	0.55	0.6
ResNet (MLP)	SN → YT	zero-shot	0.981	0.938	0.659	0.658	0.231
	SN → YT	fine-tuning	0.953	0.889	0.672	0.670	0.377
	SN → YT	zero-shot	0.956	0.886	0.591	0.248	0.139
TrajSV	SN → YT	fine-tuning	0.980	0.931	0.719	0.723	0.386
	YT → YT	training	<b>0.984</b>	<b>0.956</b>	<b>0.791</b>	<b>0.802</b>	<b>0.475</b>

TABLE X  
RETRIEVAL SCALABILITY ON YOUTUBE.

Database size	500	1,000	1,500	2,000	2,500	3,000
Retrieval time (s)	10.70	11.87	12.58	13.03	13.44	13.75

offs. 3) Variable-length actions, comprising short-term actions like clearance (typically lasting several seconds) and long-term actions like substitution (which can span several minutes). For reference, we also include the performance of X-CLIP (MLP),

TABLE XI  
COMPARISON WITH GENERAL VIDEO MODELS ON SOCCERNET.

Components	HR@1	MRR
TrajSV	<b>0.250</b>	<b>0.551</b>
w/o Trajectory	0.208	0.489
StreamMind	0.223	0.514

the strongest baseline on SoccerNet. We observe that TrajSV consistently outperforms X-CLIP (MLP) by approximately 39.98% across diverse sports scenarios. This performance gain is consistent with the overall results reported in Table II, and is primarily attributed to TrajSV's ability to capture fine-grained trajectory patterns that are critical for identifying similar videos in retrieval tasks. This is further illustrated in the case study of the corner event in Figure 4, where the player movement patterns retrieved by TrajSV closely resemble those in the query video.



- (1) **Ground truth** {38:40}[1.0]: PLAYER TEAM is awarded a yellow card for his tackle. He doesn't seem to agree with the decision but REFEREE ignores the protests.
- (2) **Baidu-VC** {38:36}[0.0]: PLAYER TEAM is booked after bringing down an opponent REFEREE had an easy decision to make.
- (3) **Baidu-VC+play2vec** {38:31}[0.0]: PLAYER TEAM is booked after bringing down an opponent REFEREE had an easy decision to make.
- (4) **Baidu-VC+TrajSV** {38:36}[1.0]: PLAYER TEAM is awarded a yellow card for his tackle. He doesn't seem to agree with the decision but REFEREE ignores the protests.

Fig. 5. Qualitative results of SDVC, where the spotting time and BLEU@4 are shown in {} and [], respectively.

TABLE XII  
EFFECTIVENESS ACROSS DIVERSE SPORTS SCENARIOS ON SOCCERNET.

Method	Concrete Events					Abstract Patterns			Variable-length Actions		
	Free-kick	Corner	Goal	Penalty	Overall	Throw-in	Kick-off	Overall	Second-level	Minute-level	Overall
X-CLIP (MLP)	0.392	0.312	0.461	0.402	0.417	0.327	0.254	0.314	0.363	0.225	0.280
TrajSV	<b>0.538</b>	<b>0.521</b>	<b>0.611</b>	<b>0.572</b>	<b>0.549</b>	<b>0.615</b>	<b>0.509</b>	<b>0.538</b>	<b>0.633</b>	<b>0.577</b>	<b>0.612</b>

## VII. CONCLUSIONS

This paper proposes TrajSV, a trajectory-based framework for learning sports video representations. The TrajSV framework consists of three components: data preprocessing, CR-Net, and VRNet. It is built on raw broadcast sports videos, making it widely applicable, and derives both video-level and clip-level representations to support various applications. Our experiments demonstrate its superior performance in sports video retrieval, action spotting, and video captioning tasks. In the future, we plan to explore more sports analytics tasks, such as player performance analysis and team tactics discovery, to further demonstrate the effectiveness of TrajSV.

## REFERENCES

- [1] Z. Wang, C. Long, G. Cong, and C. Ju, “Effective and efficient sports play retrieval with deep representation learning,” in *KDD*. ACM, 2019, pp. 499–509.
- [2] L. Sha, P. Lucey, Y. Yue, P. Carr, C. Rohlf, and I. A. Matthews, “Chalkboarding: A new spatiotemporal query paradigm for sports play retrieval,” in *IUI*. ACM, 2016, pp. 336–347.
- [3] Q. Zhang, Z. Wang, C. Long, and S. Yiu, “On predicting and generating a good break shot in billiards sports,” in *SDM*. SIAM, 2022, pp. 109–117.
- [4] T. Decroos, J. V. Haaren, and J. Davis, “Automatic discovery of tactics in spatio-temporal soccer match data,” in *KDD*. ACM, 2018, pp. 223–232.
- [5] M. Di, D. Klabjan, L. Sha, and P. Lucey, “Large-scale adversarial sports play retrieval with learning to rank,” *ACM Trans. Knowl. Discov. Data*, vol. 12, no. 6, pp. 69:1–69:18, 2018.
- [6] T. Haruyama, S. Takahashi, T. Ogawa, and M. Haseyama, “Similar scene retrieval in soccer videos with weak annotations by multimodal use of bidirectional LSTM,” in *MMSAsia*. ACM, 2020, pp. 27:1–27:8.
- [7] L. Probst, F. Rauschenbach, H. Schuldt, P. Seidenschwarz, and M. Rumo, “Integrated real-time data stream analysis and sketch-based video retrieval in team sports,” in *IEEE BigData*. IEEE, 2018, pp. 548–555.
- [8] I. A. Kabary and H. Schuldt, “Sportsense: using motion queries to find scenes in sports videos,” in *CIKM*. ACM, 2013, pp. 2489–2492.
- [9] A. Cioppa, A. Deliège, S. Giancola, B. Ghanem, M. V. Droogenbroeck, R. Gade, and T. B. Moeslund, “A context-aware loss function for action spotting in soccer videos,” in *CVPR*. Computer Vision Foundation / IEEE, 2020, pp. 13 123–13 133.
- [10] T. Haruyama, S. Takahashi, T. Ogawa, and M. Haseyama, “Retrieval of similar scenes based on multimodal distance metric learning in soccer videos,” in *MMSports@MM*. ACM, 2019, pp. 10–15.
- [11] L. Kong, D. Huang, J. Qin, and Y. Wang, “A joint framework for athlete tracking and action recognition in sports videos,” *IEEE TCSVT*, vol. 30, no. 2, pp. 532–548, 2019.
- [12] H. Mkhallati, A. Cioppa, S. Giancola, B. Ghanem, and M. Van Droogenbroeck, “SoccerNet-Caption: Dense video captioning for soccer broadcasts commentaries,” in *CVPRW CVsports*, 2023.
- [13] M. Qi, Y. Wang, A. Li, and J. Luo, “Sports video captioning via attentive motion representation and group relationship modeling,” *IEEE TCSVT*, vol. 30, no. 8, pp. 2617–2633, 2019.
- [14] Z. Wang, C. Long, and G. Cong, “Similar sports play retrieval with deep reinforcement learning,” *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 4, pp. 4253–4266, 2023.
- [15] “Stats perform,” <https://www.stats.com/artificial-intelligence>.
- [16] Z. Tong, Y. Song, J. Wang, and L. Wang, “Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training,” *CoRR*, vol. abs/2203.12602, 2022.
- [17] B. Ni, H. Peng, M. Chen, S. Zhang, G. Meng, J. Fu, S. Xiang, and H. Ling, “Expanding language-image pretrained models for general video recognition,” in *ECCV*, 2022, pp. 1–18.
- [18] Q. Zhang, Z. Wang, C. Long, and S.-M. Yiu, “Billiards sports analytics: Datasets and tasks,” *TKDD*, 2025.
- [19] Y. Xu and Y. Fu, “Sports-traj: A unified trajectory generation model for multi-agent movement in sports,” in *ICLR*, 2025.
- [20] A. Gu and T. Dao, “Mamba: Linear-time sequence modeling with selective state spaces,” *arXiv preprint arXiv:2312.00752*, 2023.
- [21] I. A. Kabary and H. Schuldt, “Enhancing sketch-based sport video retrieval by suggesting relevant motion paths,” in *SIGIR*. ACM, 2014, pp. 1227–1230.
- [22] S. Giancola and B. Ghanem, “Temporally-aware feature pooling for action spotting in soccer broadcasts,” in *CVPR Workshops*. Computer Vision Foundation / IEEE, 2021, pp. 4490–4499.
- [23] M. Tomei, L. Baraldi, S. Calderara, S. Bronzin, and R. Cucchiara, “Rmsnet: Regression and masking for soccer event spotting,” in *ICPR*. IEEE, 2020, pp. 7699–7706.
- [24] X. Zhou, L. Kang, Z. Cheng, B. He, and J. Xin, “Feature combination meets attention: Baidu soccer embeddings and transformer based temporal detection,” *CoRR*, vol. abs/2106.14447, 2021.
- [25] F. Wu, Q. Wang, J. Bian, N. Ding, F. Lu, J. Cheng, D. Dou, and H. Xiong, “A survey on video action recognition in sports: Datasets, methods and applications,” *IEEE Trans. Multim.*, vol. 25, pp. 7943–7966, 2023.
- [26] J. Li, L. Zhang, Q. Wu, Z. Qi, H. Lu, M. Wang, and D. Tao, “Contextualized relation predictive model for self-supervised group activity representation learning,” *IEEE TMM*, vol. 26, pp. 354–368, 2024.
- [27] Z. Wang, Z. Li, X. Lang, Y. Zheng, M. Tian, L. Wu, L. Wang, and C. Chen, “Knowledge augmented relation inference for group activity recognition,” *IEEE TCSVT*, pp. 11 645–11 658, 2024.
- [28] Y. Hu, B. Han, G. Wang, and X. Lin, “Enhanced shot change detection using motion features for soccer video analysis,” in *ICME*. IEEE Computer Society, 2007, pp. 1555–1558.
- [29] J. Wang, E. Chng, and C. Xu, “Soccer replay detection using scene transition structure analysis,” in *ICASSP (2)*. IEEE, 2005, pp. 433–436.
- [30] J. Theiner and R. Ewerth, “Tvcilib: Camera calibration for sports field registration in soccer,” in *WACV*. IEEE, 2023, pp. 1166–1175.
- [31] D. Farin, S. Krabbe, P. H. N. de With, and W. Effelsberg, “Robust camera calibration for sport videos using court models,” in *Storage and Retrieval*

- Methods and Applications for Multimedia*, ser. SPIE Proceedings, vol. 5307. SPIE, 2004, pp. 80–91.
- [32] W. Lu, J. Ting, J. J. Little, and K. P. Murphy, “Learning to track and identify players from broadcast sports videos,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 7, pp. 1704–1716, 2013.
- [33] J. Sullivan and S. Carlsson, “Tracking and labelling of interacting multiple targets,” in *ECCV (3)*, ser. Lecture Notes in Computer Science, vol. 3953. Springer, 2006, pp. 619–632.
- [34] H. Kim, C. J. Kim, M. Jeong, J. Lee, J. Yoon, and S.-K. Ko, “Cost-efficient and bias-robust sports player tracking by integrating gps and video,” in *International Workshop on Machine Learning and Data Mining for Sports Analytics*. Springer, 2022, pp. 74–86.
- [35] A. Tejero-de-Pablos, Y. Nakashima, T. Sato, N. Yokoya, M. Linna, and E. Rahtu, “Summarization of user-generated sports video by using deep action recognition features,” *IEEE Trans. Multim.*, vol. 20, no. 8, pp. 2000–2011, 2018.
- [36] A. Delière, A. Cioppa, S. Giancola, M. J. Seikavandi, J. V. Dueholm, K. Nasrollahi, B. Ghanem, T. B. Moeslund, and M. V. Droogenbroeck, “Soccernet-v2: A dataset and benchmarks for holistic understanding of broadcast soccer videos,” in *CVPR Workshops*. Computer Vision Foundation / IEEE, 2021, pp. 4508–4519.
- [37] S. Giancola, M. Amine, T. Dghaily, and B. Ghanem, “Soccernet: A scalable dataset for action spotting in soccer videos,” in *CVPR Workshops*. Computer Vision Foundation / IEEE Computer Society, 2018, pp. 1711–1721.
- [38] H.-C. Shih, “A survey of content-aware video analysis for sports,” *IEEE Transactions on circuits and systems for video technology*, vol. 28, no. 5, pp. 1212–1231, 2017.
- [39] T. Han, W. Xie, and A. Zisserman, “Video representation learning by dense predictive coding,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2019, pp. 0–0.
- [40] C. Sun, A. Myers, C. Vondrick, K. Murphy, and C. Schmid, “Videobert: A joint model for video and language representation learning,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 7464–7473.
- [41] A. Piergiovanni, A. Angelova, and M. S. Ryoo, “Evolving losses for unsupervised video representation learning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 133–142.
- [42] J. Yang, Y. Bisk, and J. Gao, “Taco: Token-aware cascade contrastive learning for video-text alignment,” in *ICCV*. IEEE, 2021, pp. 11542–11552.
- [43] H. Luo, L. Ji, M. Zhong, Y. Chen, W. Lei, N. Duan, and T. Li, “Clip4clip: An empirical study of CLIP for end to end video clip retrieval and captioning,” *Neurocomputing*, vol. 508, pp. 293–304, 2022.
- [44] S. Guo, Z. Xiong, Y. Zhong, L. Wang, X. Guo, B. Han, and W. Huang, “Cross-architecture self-supervised video representation learning,” in *CVPR*. IEEE, 2022, pp. 19 248–19 257.
- [45] R. Qian, T. Meng, B. Gong, M. Yang, H. Wang, S. J. Belongie, and Y. Cui, “Spatiotemporal contrastive video representation learning,” in *CVPR*. Computer Vision Foundation / IEEE, 2021, pp. 6964–6974.
- [46] L. Chen, G. Papandreou, F. Schroff, and H. Adam, “Rethinking atrous convolution for semantic image segmentation,” *CoRR*, vol. abs/1706.05587, 2017.
- [47] Y. Zhang, C. Wang, X. Wang, W. Zeng, and W. Liu, “Fairmot: On the fairness of detection and re-identification in multiple object tracking,” *Int. J. Comput. Vis.*, vol. 129, no. 11, pp. 3069–3087, 2021.
- [48] X. Zhou, D. Wang, and P. Krähenbühl, “Objects as points,” *CoRR*, vol. abs/1904.07850, 2019.
- [49] J. Devlin, M. Chang, K. Lee, and K. Toutanova, “BERT: pre-training of deep bidirectional transformers for language understanding,” in *NAACL-HLT (1)*. Association for Computational Linguistics, 2019, pp. 4171–4186.
- [50] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” in *NIPS*, 2017, pp. 5998–6008.
- [51] L. J. Ba, J. R. Kiros, and G. E. Hinton, “Layer normalization,” *CoRR*, vol. abs/1607.06450, 2016.
- [52] Y. A. Malkov and D. A. Yashunin, “Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 4, pp. 824–836, 2020.
- [53] D. H. Ballard, *An introduction to natural computation*, ser. Complex adaptive systems. MIT Press, 2000.
- [54] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [55] Y. Cui, C. Zeng, X. Zhao, Y. Yang, G. Wu, and L. Wang, “Sportsmot: A large multi-object tracking dataset in multiple sports scenes,” *arXiv preprint arXiv:2304.05170*, 2023.
- [56] A. Cioppa, S. Giancola, V. Somers, F. Magera, X. Zhou, H. Mkhallati, A. Delière, J. Held, C. Hinostroza, A. M. Mansourian *et al.*, “Soccernet 2023 challenges results,” *arXiv preprint arXiv:2309.06006*, 2023.
- [57] P. Soille, *Morphological image analysis: principles and applications*. Springer Science & Business Media, 2013.
- [58] I. Loshchilov and F. Hutter, “Decoupled weight decay regularization,” *arXiv preprint arXiv:1711.05101*, 2017.
- [59] L. N. Smith and N. Topin, “Super-convergence: Very fast training of neural networks using large learning rates,” in *Artificial intelligence and machine learning for multi-domain operations applications*, vol. 11006. SPIE, 2019, pp. 369–386.
- [60] A. Cioppa, S. Giancola, A. Delière, L. Kang, X. Zhou, Z. Cheng, B. Ghanem, and M. V. Droogenbroeck, “Soccernet-tracking: Multiple object tracking dataset and benchmark in soccer videos,” in *CVPR Workshops*, 2022.
- [61] X. Zhou, D. Wang, and P. Krähenbühl, “Objects as points,” *arXiv preprint arXiv:1904.07850*, 2019.
- [62] D. P. Kingma, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.
- [63] Y. Ma, G. Xu, X. Sun, M. Yan, J. Zhang, and R. Ji, “X-CLIP: end-to-end multi-grained contrastive learning for video-text retrieval,” in *ACM Multimedia*. ACM, 2022, pp. 638–647.
- [64] Y. Liu, K. Zhao, and G. Cong, “Efficient similar region search with deep metric learning,” in *KDD*. ACM, 2018, pp. 1850–1859.
- [65] X. Ding, H. Wu, Y. Yang, S. Jiang, D. Bai, Z. Chen, and T. Cao, “Streammind: Unlocking full frame rate streaming video dialogue through event-gated cognition,” *ICCV*, 2025.



**Zheng Wang** is currently a Principal Researcher and Huawei TopMinds at Huawei Technologies, Co., Ltd.. His current research interest focuses on multimodal search. He received his PhD degree at Nanyang Technological University. His research has been recognized by some prestigious awards, including Rising Star Award in Spatial Data Intelligence from ACM SIGSPATIAL China, one of the Best PhD Thesis Awards, WAIC Yunfan Award, Nominated Schmidt Science Fellows, Google PhD Fellowship, and AISG PhD Fellowship.



**Shihao Xu** is a Research Scientist at Huawei Technologies, Co., Ltd., multimodal search and recommendation lab. His current research interests and works fill in multimodal applications including sports video representations, user intention generation, multimodal geometry problem solving, and multimodal prompting. He got his PhD degree in Nanyang Technological University at 2022, during which, he was working on the audio-visual understanding of human behaviours.



**Wei Shi** is currently head of multimodal search team at Huawei Technologies, Co., Ltd.. He received his PhD degree at Department of Computer Science and Technology, Tsinghua University in 2015. His research interests are broadly in multimodal search, vision-language alignment, and big data systems.